

110360129 林軒至 應用軟體實習

讀書會

組員:

110360121 廖華翊

110360129 林軒至

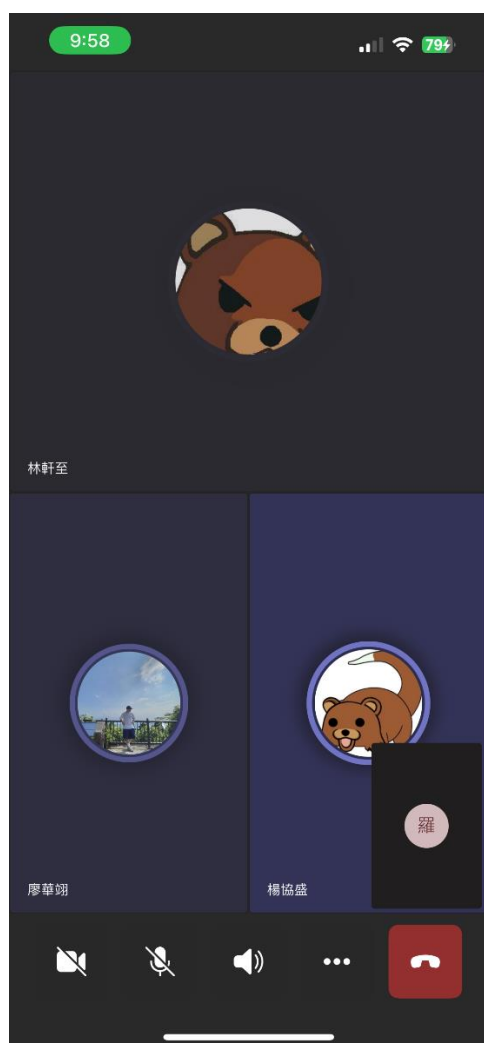
110360130 楊協盛

110360155 羅寶娜

討論時間: 2023/11/30 21:30-22:00

地點: Teams 線上開會 Line 討論

開會截圖:



<https://github.com/s098031/112-1-mmslab-java-ch04>

問 ChatGPT 問題

1. `Car` 類別：

- `protected int num` 和 `protected double gas` 是 `Car` 類別的屬性，它們被設置為 `protected`，這表示它們可以在子類別中訪問。
- `public Car()` 是 `Car` 類別的構造函數，它在創建 `Car` 物件時被呼叫，並初始化 `num`



You

```
C:\Users\chest> cmd /C ""C:\Program Files\Java\jdk-17.0.1\bin\java.exe" -
agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:1717 -
XX:+ShowCodeDetailsInExceptionMessages -cp
C:\Users\chest\AppData\Local\Temp\vscode\ws_08568\jdt_ws\jdt.ls-java-
project\bin hw7 "
生產了車子
將車號設為1234,汽油量設為20.5
Car@548c4f57
```



ChatGPT

您的程式碼看起來是正確的，並且顯示了預期的輸出。但是，當您使用 `System.out.println(car1);` 時，它會調用 `Car` 類中的 `toString` 方法。默認情況下，`Object` 類中的 `toString` 方法返回一個字符串，格式為 `類名@哈希碼`。

如果您想要自定義 `Car` 類的 `toString` 方法，以便以您自己的方式格式化輸出，您已經在 `Car` 類中進行了正確的操作。但是，由於您在 `Car` 類中沒有提供一個特定的 `toString` 方法，它默認使用了 `Object` 類中的版本。

為了解決這個問題，您可以在 `Car` 類中覆蓋 `toString` 方法，如您已經做的那樣，但確保您的 `toString` 方法返回您期望的格式。例如：

```
java Copy code

@Override
public String toString() {
    return "車號:" + num + ";汽油量:" + gas;
}
```

這樣，當您使用 `System.out.println(car1);` 時，它應該顯示您自定義格式的輸出。



You





You

在重啟檔案後並執行除錯，就沒有亂碼了，請問這是為何



ChatGPT

這種情況可能是由於某些 IDE 環境的快取或編譯問題導致的。當您修改代碼並重新啟動檔案後，IDE 可能會重新編譯代碼並更新運行環境，這有助於解決一些臨時性的顯示問題。

一些可能的原因包括：

1. **IDE快取：** IDE 可能會在某些情況下保留先前的編譯結果或快取，當您重新啟動檔案時，它會重新編譯代碼，這可能解決顯示問題。
2. **編碼快取：** IDE 或編譯器有時會保留先前的編碼設置。重新啟動檔案後，可能會重新檢測編碼並應用正確的設置。
3. **編譯問題：** 在某些情況下，IDE 可能沒有正確地檢測到代碼的更改，或者正在使用過時的編譯結果。重新啟動檔案可能會觸發重新編譯，解決這些問題。

如果您的代碼在重新啟動後運行正常，那麼可能是由於上述原因之一。如果將來再次遇到類似的顯示問題，您可以嘗試先重新啟動 IDE，或者手動觸發一次重新編譯。





You

能在JAVA程式碼中設定終端機的編碼方式嗎？

例如 chcp 65001



ChatGPT

在 Java 程式碼中，您無法直接設定終端機的編碼方式。Java 是一種跨平台的語言，它主要依賴於 Java 虛擬機 (JVM) 來執行程式碼，而不直接控制終端機。終端機的編碼通常由操作系統的設置來控制。

如果您的目標是確保 Java 程式碼在特定環境中以特定的編碼顯示正確的輸出，您可以嘗試以下方法：

1. 在程式碼中設定編碼：在 Java 程式碼中，您可以使用 `System.setProperty` 來設定默認的編碼。例如：

java

Copy code

```
System.setProperty("file.encoding", "UTF-8");
```

這樣做會影響 `System.out.println` 等輸出。請注意，這僅影響 Java 程式碼的內部編碼，並不直接影響終端機編碼。

2. 使用 `PrintWriter`：如果您將輸出重定向到文件或終端機，可以使用 `PrintWriter`，它允許您指定編碼。例如：

java

Copy code

```
PrintWriter writer = new PrintWriter(System.out, true, "UTF-8");  
writer.println("您的輸出");
```

這樣做可以確保在輸出時使用指定的編碼。

3. 確保終端機支援 UTF-8：在使用終端機時，確保終端機的編碼設置為 UTF-8。這通常可以在終端機的屬性或設置中進行設定。

請注意，這些方法的效果可能因操作系統和終端機而異。在某些情況下，可能需要使用特定的工具或命令，如 `chcp`（在 Windows 中）或類似的命令。

最後，確保您的編碼設定不會對其他程式碼造成負面影響，並謹慎使用此類設定，以避免意外的行為。

👍 🧡 🤖 🔄

心得:

哈...結果到最後亂碼的問題莫名其妙在重開檔案後就解決了，難道這是東方神秘力量？